

Urbanisation et BPM
Retours d'expérience sur le SI de Bouygues Telecom

26 Janvier 2006
Jeudis de l'objet
Yves Caseau

Plan

- I: Urbanisation & Bouygues Telecom
- II: BPM & Architecture
- III: BPM & Qualité des données
- IV: BPM & Exploitation



Première Partie: BPM & Bouygues Telecom

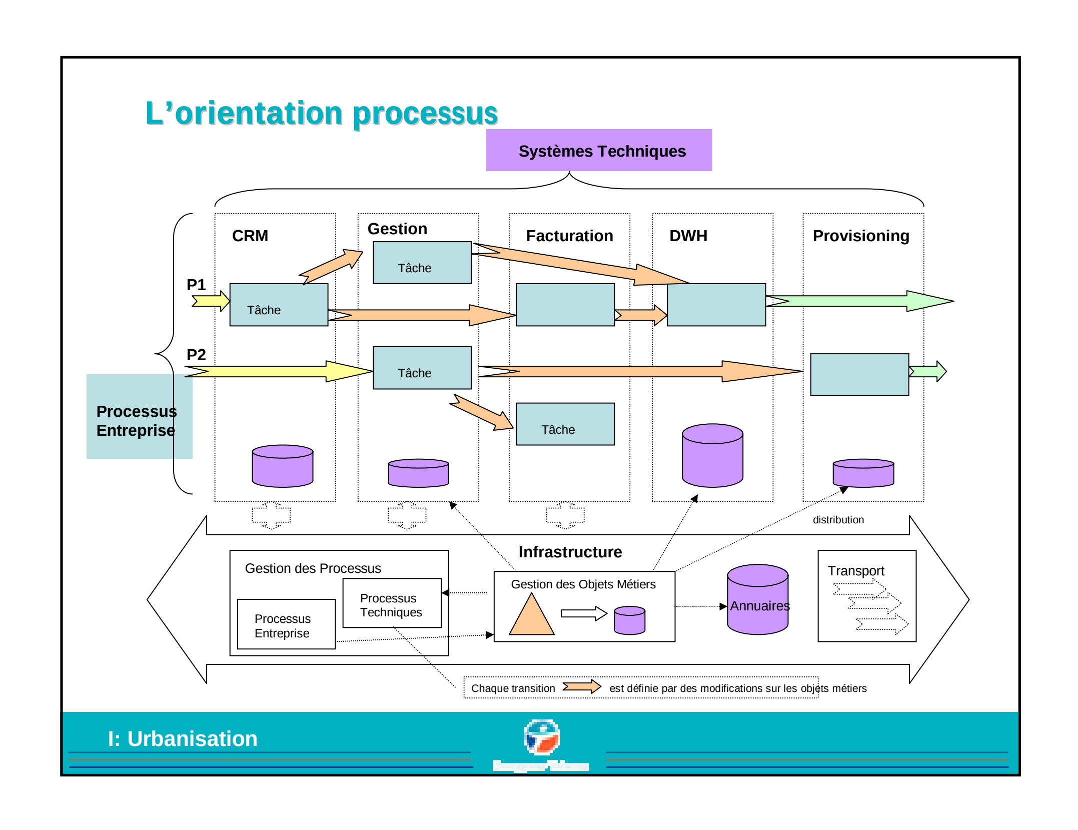
- Refonte du SI de Bouygues Telecom
- L'orientation processus
- Les Objets Métier
- Les Processus Métier
- Urbanisation du « back-office »

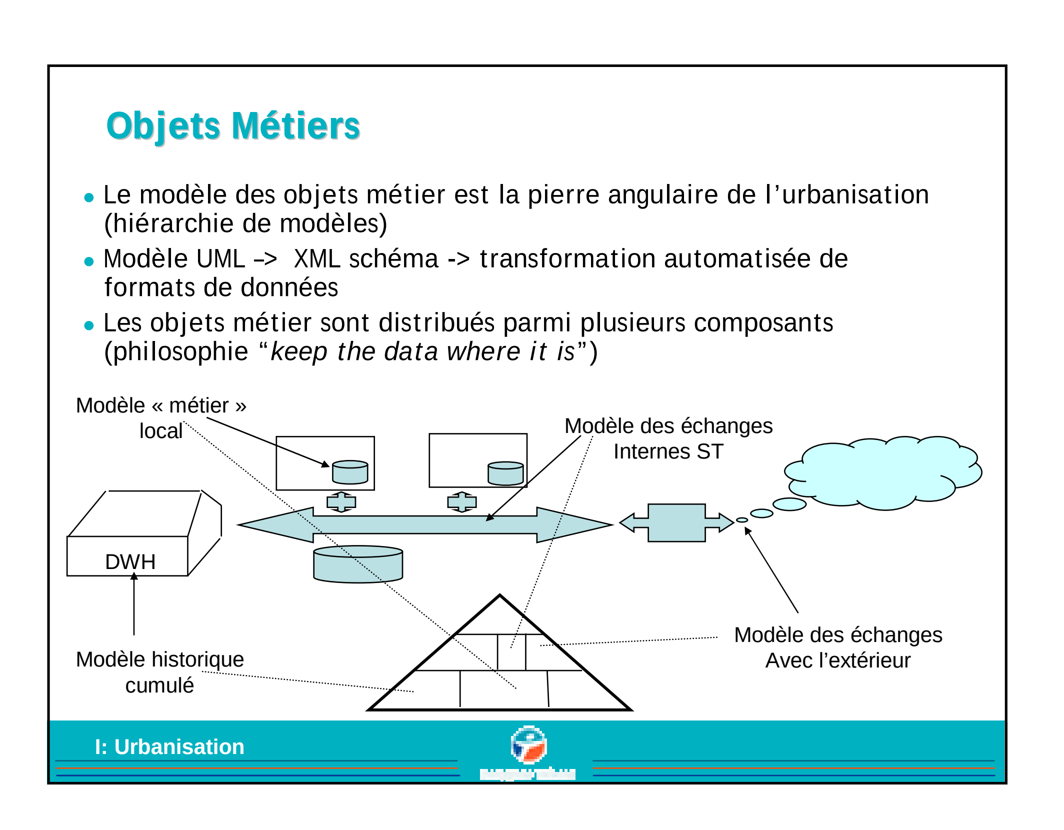


Histoire du SI de Bouygues Telecom

- 95-99 (croissance exponentielle): le SI est construit autour du progiciel BSCS (valorisation, facturation, CRM, Provisioning, gestion client, …)
- Pourquoi changer ?
  - Problèmes de capacité et de performance
  - Trop de développements ad-hoc (coûts)
  - Augmentation du "Time-to-market" et décroissance de la flexibilité
- Stratégie d'urbanisation décidée en 99-2000 :
  - Se réapproprier le SI: Objets métiers et Processus métiers, maîtrise de l'intégration
  - Performance and scalabilité: architecture à composants
  - Flexibilité: orientation processus (moteur de processflow) & components flexibles (meta-data)
  - Qualité de service: sécurisation infrastructure, monitoring SLA




L'orientation processus
Systèmes Techniques
CRM
Gestion
Facturation
DWH
Provisioning
P1
P2
Tâche
Tâche
Tâche
Tâche
Tâche
Processus Entreprise
distribution
Infrastructure
Gestion des Processus
Processus Techniques
Processus Entreprise
Gestion des Objets Métiers
Annuaires
Transport
Chaque transition est définie par des modifications sur les objets métiers
I: Urbanisation


Objets Métiers
• Le modèle des objets métier est la pierre angulaire de l'urbanisation (hiérarchie de modèles)
• Modèle UML –> XML schéma -> transformation automatisée de formats de données
• Les objets métier sont distribués parmi plusieurs composants (philosophie "keep the data where it is")
Modèle « métier » local
Modèle des échanges Internes ST
DWH
Modèle historique cumulé
Modèle des échanges Avec l'extérieur
I: Urbanisation

Urbanisation du Back-office

- Infrastructure d'intégration (WebMethods) and intégration de plates-formes de service (CORBA Visibroker): 2001-2002
- Médiation: 2001-2003
- Roaming: 1Q04 (réutilisation moteur de valorisation)
- CRM : 2001 à 2004 (Siebel 7.5)
- Provisioning: 1Q04 (développement spécifique sur base Kabira - ObjectSwitch)
- Valorisation : (spécifique) partie données 2002, complet Juillet 2004
- Facturation : 2005 – Geneva (package)



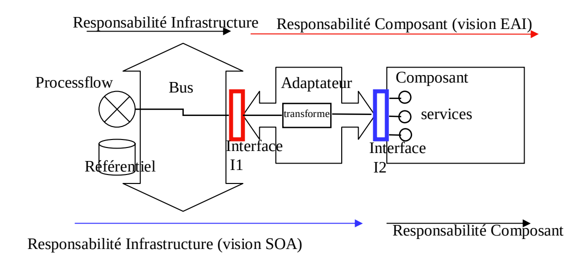
Deuxième Partie: BPM & Architecture

- Trois dimensions
- Architecture Fractale
- Cible Bouygues Telecom
- EAI & SOA réconciliés
- Agilité ?



Services et Événements

- « Services métiers » vs. « Services logiciels »

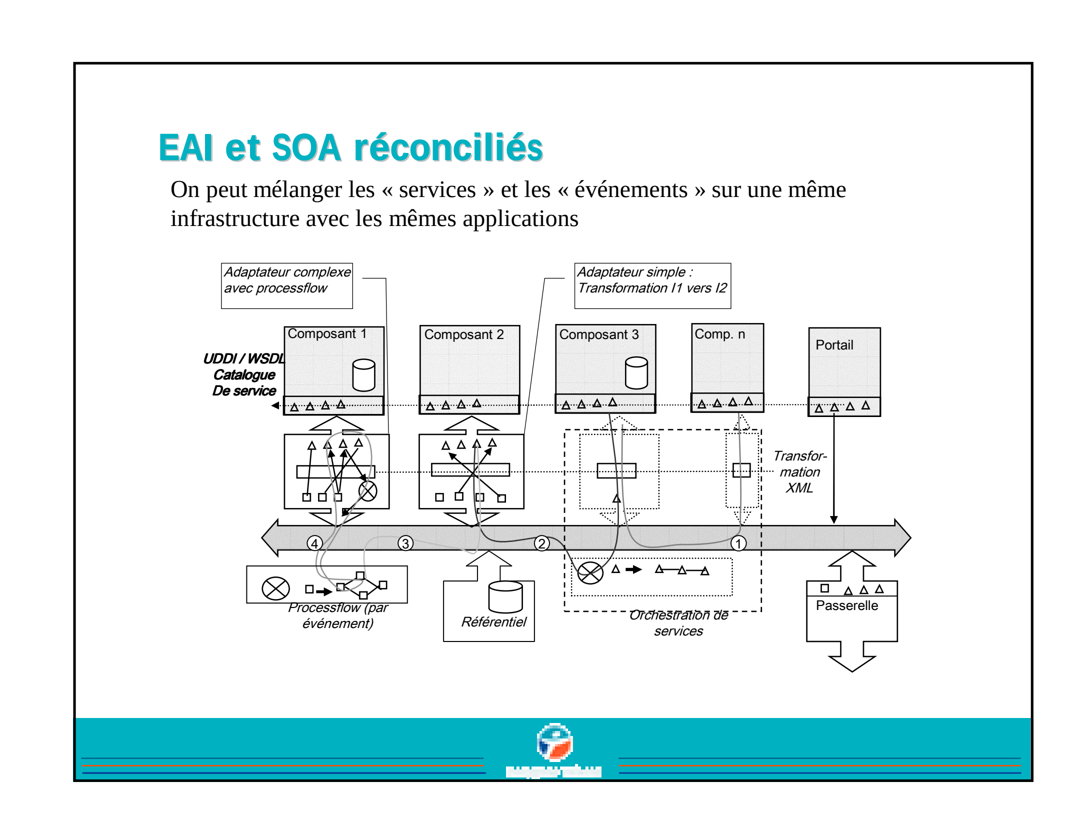


- Services métiers (plus générique, plus réutilisables) : investissement pour le futur
- Service vs. Événements: ne pas connaître son consommateur
  - Pas une question de technologie: pub/sub se traduit en orchestration de service
  - Une question de modélisation: un événement est plus réutilisable



EAI et SOA réconciliés

On peut mélanger les « services » et les « événements » sur une même infrastructure avec les mêmes applications

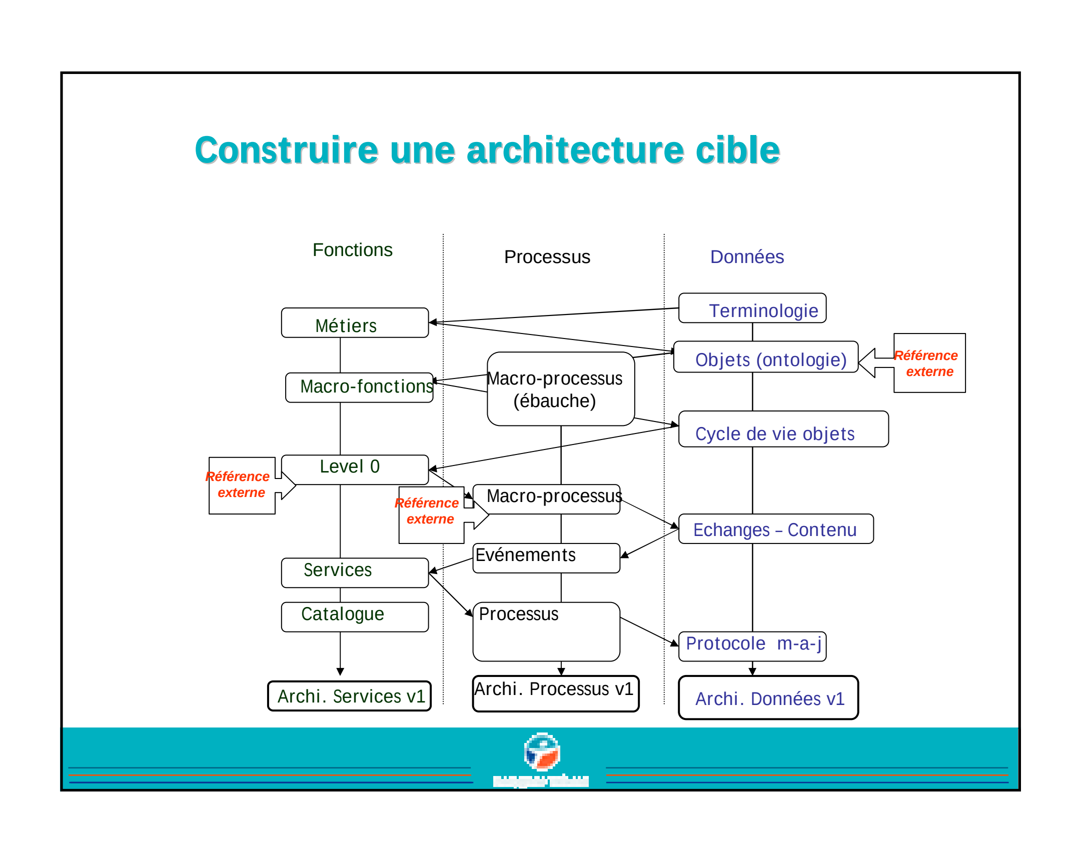


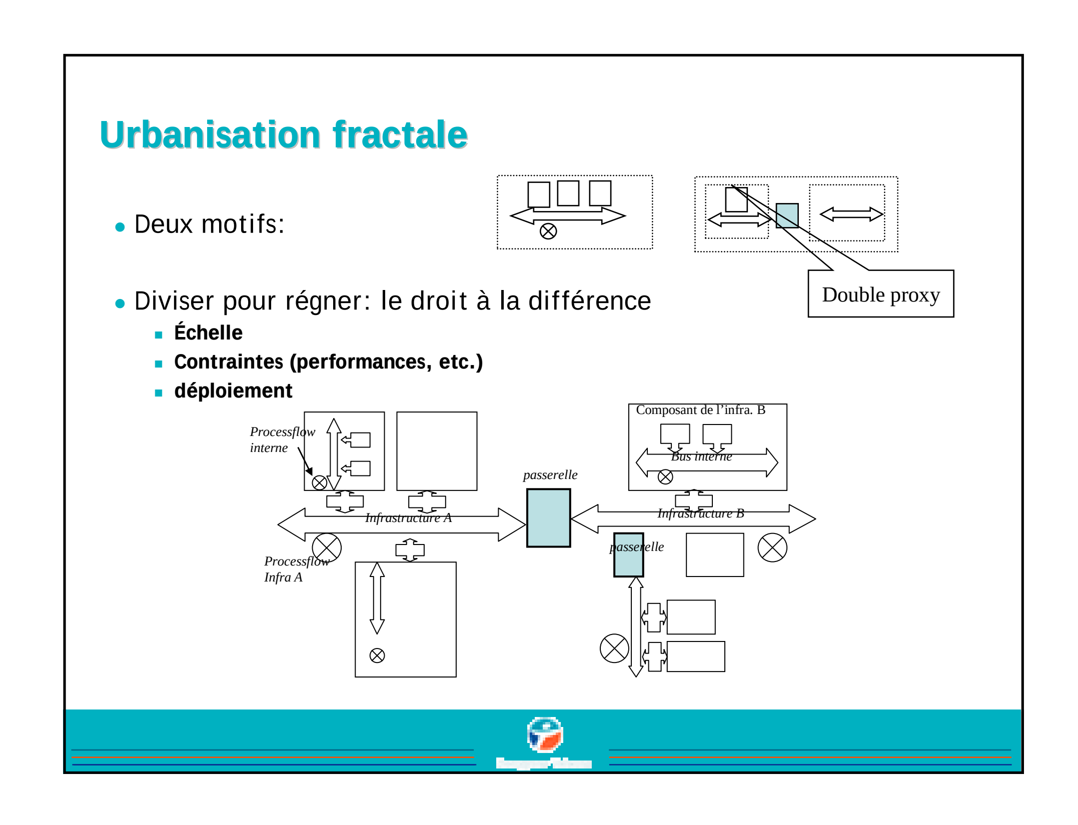
Trois dimensions de l'urbanisation

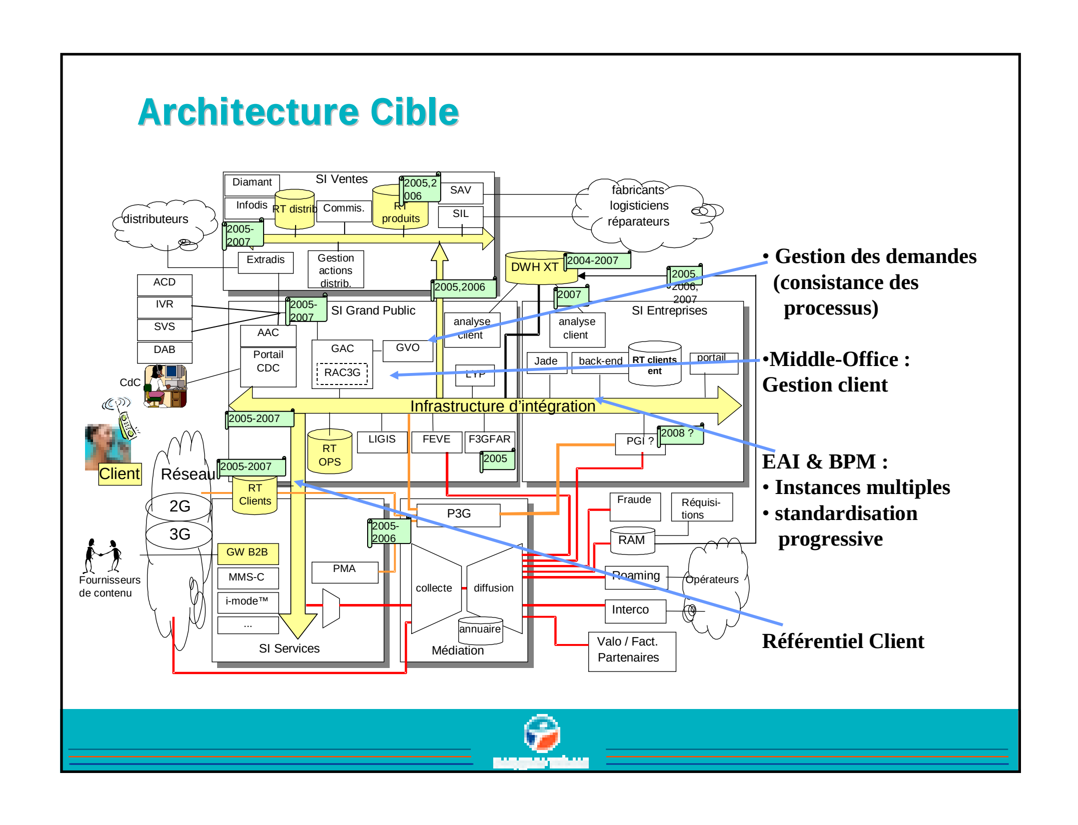
	Vision Données	Vision Services	Vision Evénements
ETL	*fournit le modèle de donnée qui est commun aux échanges*	Le modèle de donnée objet est enrichi par la vision service	permet de valider la cinématique et planification des échanges de données
SOA	induit une partie des services métiers structure les interfaces de service	*Fournit le catalogue de services qui sont implémentés par les composants*	permet de produire un ensemble d'interfaces de service qui sont facilement re-composables
EAI	les flux d'échanges de données doivent être harmonisés avec les flux de contrôle	permet de stabiliser la contribution de chaque composant au système d'information	*fournit la structure asynchrone des échanges qui permet de définir des processus*



Construire une architecture cible




Urbanisation fractale
Deux motifs:
Double proxy
Diviser pour régner: le droit à la différence
Échelle
Contraintes (performances, etc.)
déploiement
Processflow interne
Composant de l'infra. B
Bus interne
passerelle
Infrastructure A
Infrastructure B
passerelle
Processflow Infra A


Architecture Cible
distributeurs
Diamant
SI Ventes
SAV
Infodis
RT distrib
Commis.
RT produits
SIL
fabricants logisticiens réparateurs
Extradis
Gestion actions distrib.
DWH XT
SI Entreprises
ACD
IVR
SVS
DAB
AAC
Portail CDC
SI Grand Public
GAC
GVO
RAC3G
analyse client
Jade
back-end
RT clients ent
portail
CdC
Infrastructure d'intégration
Client
Réseau
2G
3G
RT OPS
LIGIS
FEVE
F3GFAR
PGI ?
RT Clients
P3G
Fraude
Réquisi-tions
RAM
GW B2B
MMS-C
i-mode™
PMA
collecte
diffusion
annuaire
Roaming
Opérateurs
Interco
Fournisseurs de contenu
SI Services
Médiation
Valo / Fact. Partenaires
• Gestion des demandes (consistance des processus)
•Middle-Office : Gestion client
EAI & BPM :
• Instances multiples
• standardisation progressive
Référentiel Client

Qui a dit « agilité » ?

- Paramétrage … mais tests de non-regression => 3mois
- Orientation-processus … mais besoin de redévelopper les adaptateurs
- Changement dans un échange entre deux systèmes … l'ensemble des composants est impacté

Anecdote Telcordia: du « *best of breed integration* » à la pré-intégration

- La flexibilité n'est pas une question de technologie
- Plutôt une question de conception et de **processus** (voire d'état d'esprit)



Conception modulaire et tests agiles

Conception

- Modularité de l'architecture fonctionnelle (importance de la vision métier – mutualisation/ réutilisation)
- Conception des échanges et compatibilité ascendante
- **Rôle de l'ingénierie XML**

Tests Agiles

- A inventer (processus, responsabilité, analyse)
- « bouchonner » … avec rigueur et conviction
- Créer des « trains » de tests agiles (planification)

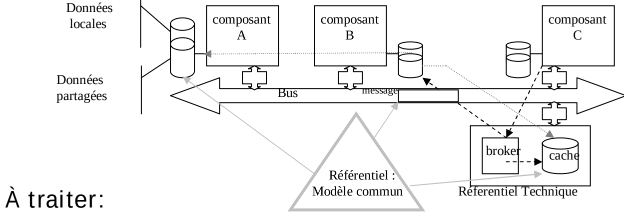


Troisième Partie: « Urbanisation et données »

- Architecture de données
- QoS et QoD
- Re-synchronisation
- Synchronisation et Transactions



Architecture de données



À traiter:

1. Synchronisation de copies
   - Gérer le flux de synchronisation
   - Garantir la cohérence des « snapshots »
     - Impossible dans le cas général
     - OK si on se limite à une cohérence modulo les observations des processus
2. Interactions
   - Les activités interagissent via (1) messages/services (2) ressources partagées (objets)
   - La cohérence => signalisation / exclusion / sérialisation



Qualité de service et qualité des données

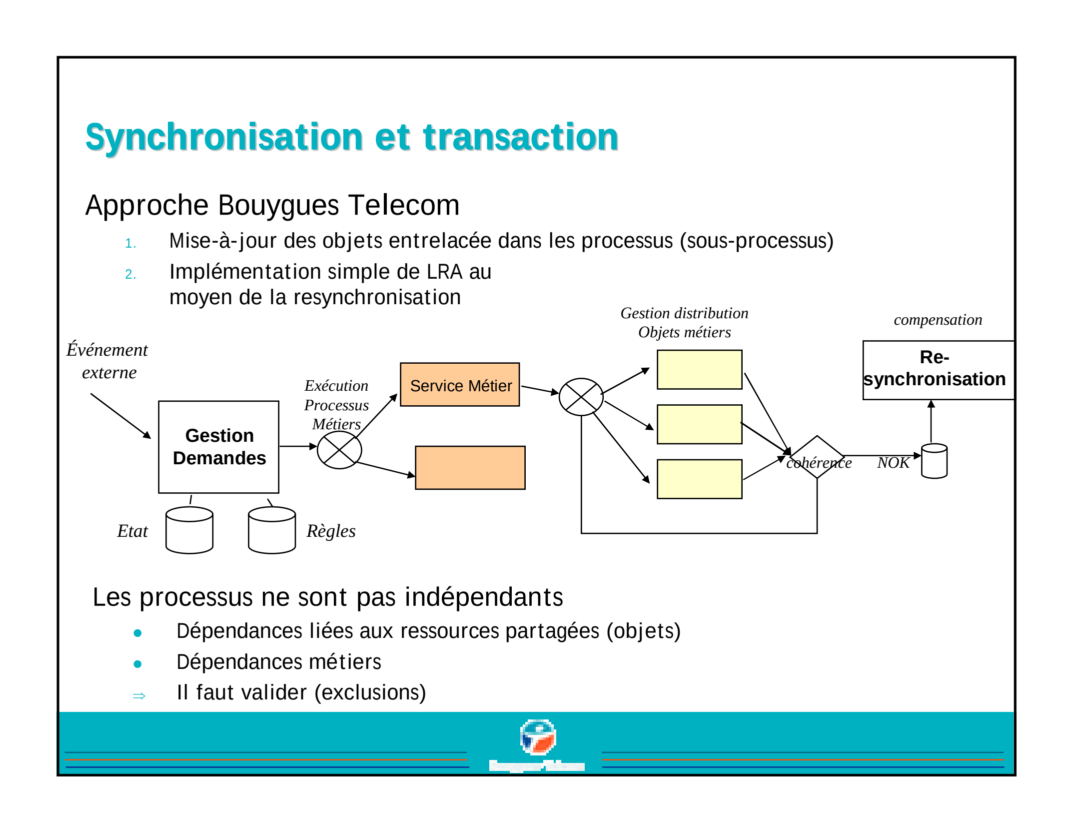
- Etudes
  - Sterling: « *Data synchronization: What is Bad Data Costing Your Company* »
  - DWHI: « *Data Quality and the bottom line – achieving business success through a commitment to high quality data* »
  - Taux d'erreurs allant de quelques % à quelques dizaines de % !
  - Impact direct : perte de revenu
- Expérience Bouygues Telecom: dégradation réciproque
  - QoS => QoD :
    - Désynchronisation => erreurs fonctionnelles
    - Baisse QoS => détournement des processus => erreurs (cohérence/saisies)
  - QoD => QoS:
    - Erreurs dans les passerelles/adaptateurs => demandes en attente
    - Temps de traitement dégradé => baisse de QoS

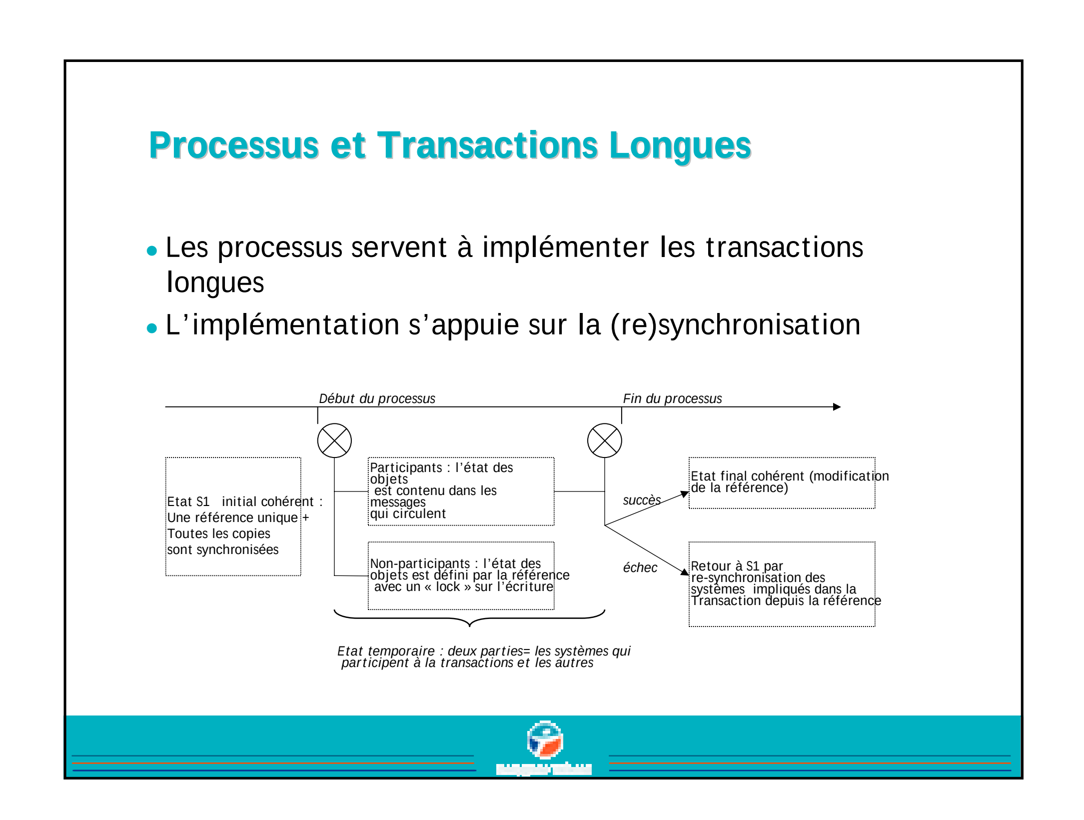


Re-synchronisation

- Désynchronisation:
  - Erreurs durant le processus
  - Crash/ incidents /reprises / retard de planification
  - Erreurs de saisie
- La désynchronisation est une condition récurrente ☺
- Besoins:
  1. Outils de re-synchronisation
     - Utilisation régulière
     - Reprise sur incident
  2. Processus permanent de nettoyage des données
     - Administration de données
     - Implication MOA (propriétaire)




Synchronisation et transaction
Approche Bouygues Telecom
1. Mise-à-jour des objets entrelacée dans les processus (sous-processus)
2. Implémentation simple de LRA au moyen de la resynchronisation
Gestion distribution Objets métiers
compensation
Événement externe
Exécution Processus Métiers
Service Métier
Re-synchronisation
Gestion Demandes
cohérence
NOK
Etat
Règles
Les processus ne sont pas indépendants
Dépendances liées aux ressources partagées (objets)
Dépendances métiers
Il faut valider (exclusions)


Processus et Transactions Longues
Les processus servent à implémenter les transactions longues
L'implémentation s'appuie sur la (re)synchronisation
Début du processus
Fin du processus
Etat S1 initial cohérent : Une référence unique + Toutes les copies sont synchronisées
Participants : l'état des objets est contenu dans les messages qui circulent
Non-participants : l'état des objets est défini par la référence avec un « lock » sur l'écriture
succès
échec
Etat final cohérent (modification de la référence)
Retour à S1 par re-synchronisation des systèmes impliqués dans la Transaction depuis la référence
Etat temporaire : deux parties= les systèmes qui participent à la transaction et les autres

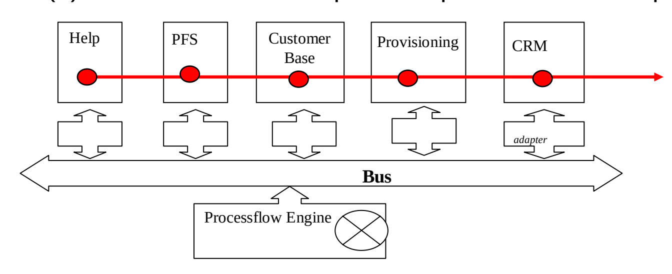
Quatrième Partie: Exploitation

- Position du problème
- Difficultés
- OAI : exemple
- Pilotage des processus
- OAI : mode d'emploi



Position du Problème

- Soit: (1) un ensemble de composants qui exécutent des processus



- (2) Un contrat de service

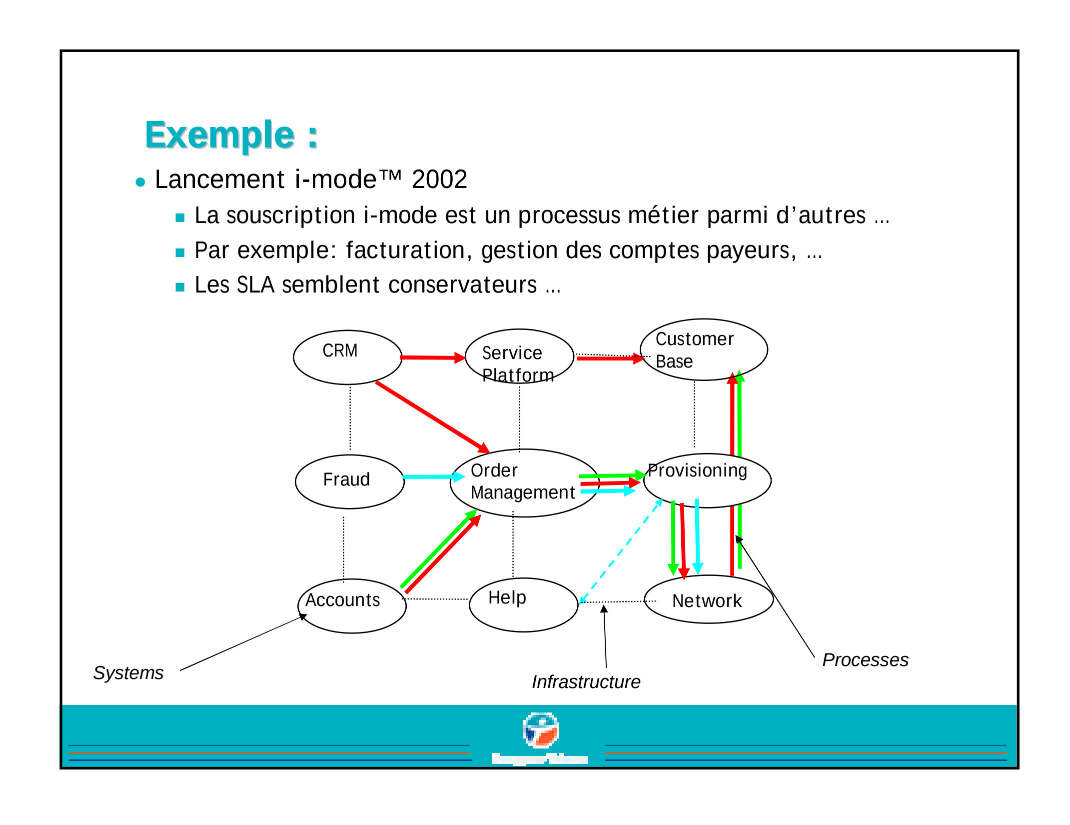
20 clients par
Heure en moins
De 2 minutes

(3) des aléas ….

- Pics d'activité
- Pannes
- Autres processus

- Question: peut-on automatiser le pilotage des processus ?





Difficultés

- Diagnostic
  - Temps réel (fil de l'eau) vs. Analyse de logs
  - Absorption de pics => détecte les problèmes trop tard
  - Capacité d'introspection à chaud
- *Capacity Planning*
  - OAI (priorités, aléas, latence, …)
  - Modèle unifié
- Planification
  - Mélange planifié / fil de l'eau
  - ! : asynchrone => accepte les pics de charge mais la QoS se dégrade => besoin de SLA explicites
- Reprise sur incident
  - À chaud -> contrainte performance
  - Ingénierie de ré-injection de messages (outils)



SLAs, Priorités et Stratégies Adaptatives

- Les processus métiers ont des priorités différentes …
  - Une stratégie adaptative devrait équilibrer la charge et les ressources pour satisfaire les SLA en fonction des priorités (“*graceful degradation*”)
  - Adaptation => doit tolérer les “bursts”
  - Adaptation => doit tolérer les indisponibilités courtes
- Deux approches possibles :
  - Ordonnancement des messages
  - Contrôle de flux
- …ont été étudiées par simulation (évènements discrets)



Ordonnancement des messages

- Tri des files d’attente : modifier l’ordre de traitement des messages
- FCFS (FIFO)
  - La méthode par défaut de la plupart des middleware (respect des contraintes temporelles)
  - Cependant, le respect de l’ordonnancement temporel est trop fort pour supporter une stratégie de distribution (nécessaire pour la fiabilité et les performances).
- LCFS (FILO)
  - Une « bonne » stratégie pour gérer les congestions.
- “SLA routing”
  - Prévision du temps de début de traitement à partir du SLA.
- Combinaison avec les priorités
  - On traite les messages à forte priorité en premier



Contrôle de flux

- [cf. Advanced Engineering Informatics 19(2005) 199-211]
- Nous avons évalué plusieurs stratégies
  - RS1: Lorsque la QoS d'un système X tombe en dessous de 90% de son SLA, nous réduisons le flux des systèmes qui fournissent X et dont la "priorité" est inférieure à celle de X.
  - RS2: Une approche similaire fondée sur les processus. Lorsque la QoS d'un processus tombe en dessous de 90%, nous réduisons le flux de tous les systèmes dont la priorité est plus faible que celle de P et qui alimentent des systèmes qui participent à P.
  - Il existe plusieurs variante sur la réduction de flux (couper, ralentir … ou changer la méthode de tri des messages).
- Le contrôle de flux est plus complexe mais n'est pas nécessairement partie du middleware d'intégration



Conclusions tirées de la simulation

Quelques pas dans la direction "*autonomic computing*"

1. *Self-optimization*:
   - La gestion des priorités fonctionne: il est possible (et assez simple) de prendre les priorités des processus en compte dans le traitement des files et d'obtenir de la sorte une véritable amélioration.
   - Les algorithmes de tri des files d'attente ont un impact fort: les méthodes sophistiqués qui utilisent la structure du SLA produisent une véritable amélioration par rapport à FCFS.
   - Les règles de contrôle de flux sont intéressantes, mais moins efficaces et plus difficiles à maîtriser.
2. *Self-healing*: nous avons obtenu une forme d' "autoréparation", mais une véritable robustesse ne peut être obtenue qu'avec une collaboration SW/HW
3. *Self-configuration*: notre objectif est de rendre la configuration déclarative (à partir des SLA) au lieu de faire de planification et ordonnancement de ressources

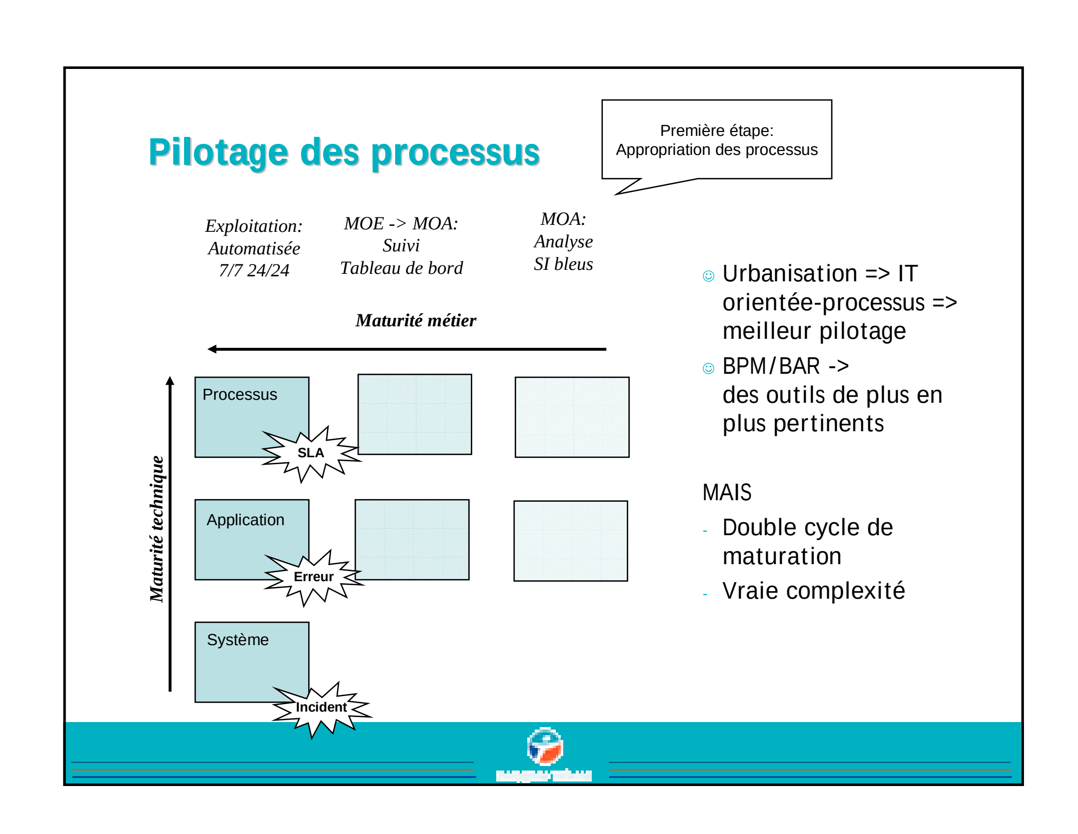


OAI: mode d'emploi

1. Conduite du changement en production
   - Formation (nouvelle culture)
   - Pilotes
   - Outils ! (manipulation de flux « vivants » - exemple: filtrage - et « morts » - réinjection -)
2. Déployer des solutions de BAM/ tests de bout-en-bout
3. Construire la prochaine génération d'outils
   - Middleware -> lobbying pour les priorités ☺
   - Simulation
   - BAM -> BAOM : pilotage actif par SLA



Pilotage des processus

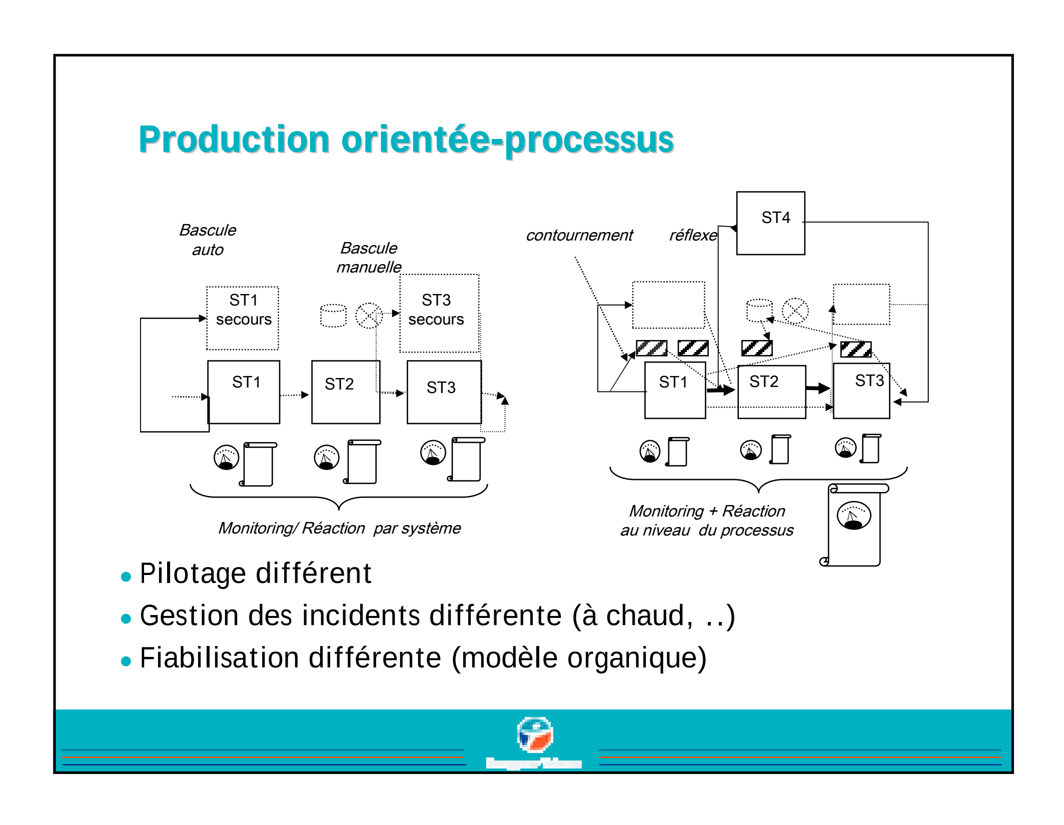


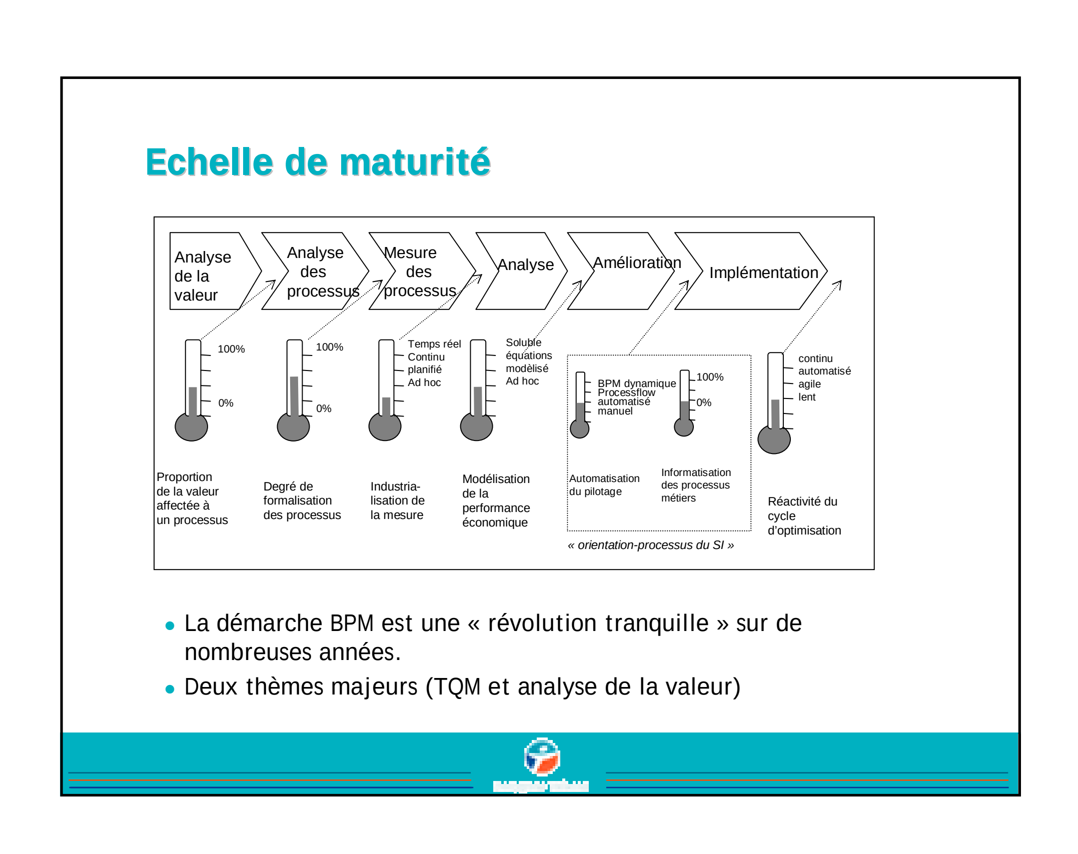
☺ Urbanisation => IT orientée-processus => meilleur pilotage

☺ BPM/BAR -> des outils de plus en plus pertinents

MAIS

- Double cycle de maturation
- Vraie complexité


Production orientée-processus
Bascule auto
Bascule manuelle
ST1 secours
ST3 secours
ST1
ST2
ST3
Monitoring/ Réaction par système
contournement
réflexe
ST4
ST1
ST2
ST3
Monitoring + Réaction au niveau du processus
Pilotage différent
Gestion des incidents différente (à chaud, ..)
Fiabilisation différente (modèle organique)


Echelle de maturité
Analyse de la valeur
Analyse des processus
Mesure des processus
Analyse
Amélioration
Implémentation
100%
0%
100%
0%
Temps réel
Continu
planifié
Ad hoc
Soluble
équations
modélisé
Ad hoc
BPM dynamique
Processflow
automatisé
manuel
100%
0%
continu
automatisé
agile
lent
Proportion de la valeur affectée à un processus
Degré de formalisation des processus
Industria-lisation de la mesure
Modélisation de la performance économique
Automatisation du pilotage
Informatisation des processus métiers
Réactivité du cycle d'optimisation
« orientation-processus du SI »
La démarche BPM est une « révolution tranquille » sur de nombreuses années.
Deux thèmes majeurs (TQM et analyse de la valeur)

Conclusion : Architecture d'Entreprise

- UML, XML, ..
  - S'appuyer sur un modèle pivot et des schémas
  - Utiliser les outils (*state-of-the-art*, ingénierie XML)
- Fractal Architecture
  - Appliquer l'approche BPM à différentes échelles
  - Construire des « régions » qui sont faiblement couplées
- Les processus ne sont pas indépendants
  => il faut implémenter une vérification de cohérence
- L'agilité n'est pas une question de technologie mais de conception
  - La modularité dépend de l'analyse fonctionnelle
  - Penser à la compatibilité ascendante des formats de message
  - Pas d'agilité sans « tests agiles »



Conclusion : Opérations et Qualité de Service

- Distribution des données => synchronisation & re-synchronisation
  - Les problèmes les plus classiques de l'informatique distribuée …
  - … mais pas les plus populaires dans le monde de l'intégration d'applications
- OAI : *Optimization of Application Integration*
  - Optimiser selon les SLA de processus et les priorités métiers
  - Un problème réellement complexe (science)
  - Besoin de progresser en terme de routage des messages et de supervision des processus
- QoS ⇔ QoD
  - La qualité de service et la qualité des données sont liées dans les deux sens
  - Attention aux migrations, audits, synchronisations, purges, …
- Cf. le livre « Urbanisation et BPM » (Dunod) ☺

URBANISATION ET BPM

